CLAIMS:

What is claimed is:

1   1.   A method of testing a device driver on a data
2   processing system, said method comprising:

3   allocating, by an operating system, a data space for
4   executing a device driver;

5   executing the device driver as an application on top
6   of the operating system to test the device driver;

7   monitoring to detect whether a request made by the
8   device driver specifies a target address within the data
9   space; and

10   in response to detecting the target address for the
11   request being made outside of the data space, trapping on
12   the target address for the request and executing a data
13   exception handler that emulates a target device.

2.   The method according to Claim 1, further comprises:

setting up, by the operating system, the data exception handler.

3.   The method according to Claim 2, wherein the setting up step further comprises:

calling the operating system to install a data exception handler facility containing the data exception handler into a vector of an interrupt table for the operating system; and

calling the data exception handler facility to register data that is used to determine when the data exception is to be taken wherein the data include each of various target addresses at which the data exception is to be taken, lengths of the target addresses, and user callback routines for calling back to the application into a data exception database table.

4.   The method according to Claim 3, further comprises:

passing, by the data exception handler facility, the data space to the vector of the interrupt table.

5.   The method according to Claim 4, wherein the data exception handler is a memory mapped input/output (IO) exception handler comprising a parser with a disassembler that is used for disassembling and identifying an assembler instruction of the request.

6.   The method according to Claim 5, wherein the assembler instruction is identified as a load command.

1   7.   The method according to Claim 5, wherein the
2   assembler instruction is identified as a store command.

1   8.   The method according to Claim 3, wherein the data
2   exception handler further comprises:

3   saving, into a memory stack, data for the request
4   which includes at least the target address for the
5   request, content data relating to the request, and a data
6   address range for the content data;

7   determining whether the target address that is saved
8   into the memory stack is within the database exception
9   database table;

10   immediately terminating the method of testing if the
11 target address is not within the database exception
12 database table;

13   disassembling the data for the request into
14 disassembled information and passing the disassembled
15 information into a respective one of the user callback
16 routines stored in the data exception database table
17 wherein the respective one of the user callback routines
18 that is used is based on the request and is for emulating
19 a behavior of the target device;

20   emulating, by the respective one of the user
21 callback routines, the behavior of the target device;

22   setting a next instruction address to an address in
23 the memory stack that is after the currently saved target
24 address; and

25   unwinding the memory stack by the data exception

26 handler to return control back to the application.

1 9. The method according to Claim 8, further comprising:

2 copying, as necessary, the content data by the
3 respective one of the user callback routines from another
4 data space for the respective one of the user callback
5 routines based on the request.

1 10. The method according to Claim 1, wherein the data
2 exception handler is a software emulator of the target
3 device and further comprises:

4 using a hardware emulator to test the application
5 for the device driver if the hardware emulator is coupled
6 to the data processing system; and

7

8 using the software emulator to test the application
9 for the device driver if the hardware emulator is not
10 coupled to the data processing system.

11. The method according to Claim 1, further comprises:

2 determining whether the application for the device
3 driver has finished executing;

4 continuing with the executing of the application if
5 the application has not finished executing; and

6 terminating the method of testing when the
7 application has finished executing.

12.   A data processing system for testing a device
driver, said data processing system comprising:

        a processor and a memory system, wherein:

        said processor executes an operating system that
allocates a data space for executing applications and
executes a device driver as an application on top of the
operating system to test the device driver; and

        said processor and said memory system, responsive to
detecting a request by the device driver specifying a
target address outside of the data space, trap on the
target address and execute a data exception handler that
emulates a target device of the device driver.

13. The data processing system according to Claim 12,
wherein said processor and said memory system call a data
exception handler facility to register data that is used
to determine when a data exception is to be taken,
wherein the data include each of various target addresses
at which the data exception is to be taken, lengths of
the target addresses, and user callback routines for
calling back to the application into a data exception
database table.

14. The system according to Claim 13, wherein said
processor and said memory system call the operating
system to install the data exception handler facility
containing the data exception handler into a vector of an
interrupt table for the operating system.

15. The system according to Claim 14, wherein the data
exception handler facility passes the data space to the
vector of the interrupt table.

16. The system according to Claim 15, wherein the data
exception handler is a memory mapped input/output (IO)
exception handler that is stored in the memory system and
comprises a parser with a disassembler that is used for
disassembling and identifying an assembler instruction of
the request.

17. The system according to Claim 16, wherein the
assembler instruction is a load command.

18. The system according to Claim 16, wherein the
assembler instruction is a store command.

19. The system according to Claim 14, wherein the data
exception handler determines whether the target address

3 is within the data exception database table and,

4 responsive to said determination, immediately terminates

5 testing of the device driver if the target address is not

6 within the data exception database table, and, emulates,

7 by the respective one of the user callback routines, the

8 behavior of the target device if the target address is

9 within the data exception database table.

1 20. The system according to Claim 19, wherein the data

2 exception handler further:

3

4   copies, as necessary, the content data by the

5 respective one of the user callback routines from another

6 data space for the respective one of the user callback

7 routines based on the request.

1 21. The system according to Claim 12, wherein the data

2 exception handler is a software emulator of the target

3 device and wherein:

4   a hardware emulator is used to test the application

5 for the device driver if the hardware emulator is coupled

6 to the data processing system; and

7   the software emulator is used to test the

8 application for the device driver if the hardware

9 emulator is not coupled to the data processing system.

10

22. A program product for testing a device driver on a
data processing system comprising:

        instruction means for allocating, by an operating
system, a data space for executing a device driver;

        instruction means for executing the device driver as
an application on top of the operating system to test the
device driver;

        instruction means for monitoring to detect whether a
request made by the device driver specifies a target
address within the data space;

        in response to detecting the target address for the
request being made outside of the data space, instruction
means for trapping on the target address for the request
and executing a data exception handler that emulates a
target device; and

        computer usable media bearing said instruction
means.

23. The program product according to Claim 22, further comprises:

instruction means for setting up, by the operating system, the data exception handler.

24. The program product according to Claim 23, wherein the instruction means for setting up further comprises:

instruction means for calling the operating system to install a data exception handler facility containing the data exception handler into a vector of an interrupt table for the operating system; and

instruction means for calling the data exception handler facility to register data that is used to determine when the data exception is to be taken wherein the data include each of various target addresses at which the data exception is to be taken, lengths of the target addresses, and user callback routines for calling back to the application into a data exception database table.

25. The program product according to Claim 24, further comprises:

instruction means for passing, by the data exception handler facility, the data space to the vector of the interrupt table.

26. The program product according to Claim 22, wherein the data exception handler further comprises:

instruction means for saving, into a memory stack,

4   data for the request which includes at least the target

5   address for the request, content data relating to the

6   request, and a data address range for the content data;

7   instruction means for determining whether the target

8   address that is saved into the memory stack is within the

9   database exception database table;

10   instruction means for immediately terminating the

11   testing of the device driver if the target address is not

12   within the database exception database table;

13   instruction means for disassembling the data for the

14   request into disassembled information and passing the

15   disassembled information into a respective one of the

16   user callback routines stored in the data exception

17   database table wherein the respective one of the user

18   callback routines that is used is based on the request

19   and is for emulating a behavior of the target device;

20   instruction means for emulating, by the respective

21   one of the user callback routines, the behavior of the

22   target device;

23   instruction means for setting a next instruction

24   address to an address in the memory stack that is after

25   the currently saved target address; and

26   instruction means for unwinding the memory stack by

27   the data exception handler to return control back to the

28   application.

1   27. The program product according to Claim 26, further

2   comprising:

3   instruction means for copying, as necessary, the
4   content data by the respective one of the user callback
5   routines from another data space for the respective one
6   of the user callback routines based on the request.

1   28.   The program product according to Claim 22, wherein
2   the data exception handler is a software emulator of the
3   target device and further comprises:

4   instruction means for using a hardware emulator to
5   test the application for the device driver if the
6   hardware emulator is coupled to the data processing
7   system; and

8   instruction means for using the software emulator to
9   test the application for the device driver if the
10   hardware emulator is not coupled to the data processing
11   system.

1   29.   The program product according to Claim 22, further
2   comprises:

3   instruction means for determining whether the
4   application for the device driver has finished executing;

5   instruction means for continuing with the executing
6   of the application if the application has not finished
7   executing; and

8   instruction means for terminating the testing of the
9   device driver when the application has finished
10   executing.